

AUTOMATED INDIVIDUALIZED LEARNING PROGRAM CREATION SYSTEM AND ASSOCIATED METHODS

Cross-Reference to Related Application

This application claims priority from copending application 09/705,153, "Automated System and Methods for Creating an Individualized Learning Program," filed November 2, 2000, the disclosure of which is hereby incorporated by reference hereinto.

Field of the Invention

The invention pertains to automated interactive training prescription systems and methods. More particularly, the invention pertains to skills management and learning management systems and methods that perform employee job and/or skills assessments. Most particularly, the invention establishes a system and method for identifying skill gaps in a user and automatically mapping a precise training regimen to the identified skill gaps.

Background of the Invention

Traditionally, organizations have focused their workforce by creating fairly narrowly defined jobs and improving skill-sets of the individuals in those jobs by using periodic training. However, as the workplace environment becomes ever more dynamic, job requirements are becoming increasingly more fluid with corresponding employee skill deficiencies that need to be identified in near real-time. Modern-day organizations require their workforce to have varying levels of knowledge and skills in the same subjects. Organizations are posturing themselves to develop accurate profiles of their employees, knowing what knowledge and skills are present within the organization, the competency

levels required for each job, where the deficiencies are, and how to provide timely skills upgrading.

The traditional manual association (i.e., linkage) of knowledge and skill-set requirements to training interventions is a resource-intensive activity that cannot provide timely responses to ever-changing job requirements. Automated tools and procedures are needed to assist organizations in prescribing the exact training interventions that are needed to satisfy employee skill deficiencies. Thus, in order to fulfill these needs, skills management systems and learning management systems must incorporate an automated system and method that can perform the analysis of employee skill deficiencies and provide a precise training regimen that will bring the employee to the desired competency level. This system and method must also be responsive to organizational skill changes/upgrading and the introduction of new/changed training programs.

Summary of the Invention

It is therefore an object of the present invention to provide a system and method that identify a gap between skills possessed by a user and those desired by an organization for the user to possess.

It is also an object to provide such a system and method that creates a training regimen to fill the identified skill gap.

It is a further object to provide such a system and method for creating an interactive learning environment that is specific to a skill desired to be learned and to a skill level already possessed by the user.

It is another object is to provide such a system and method for maintaining and upgrading the linkages between skilling objects, which comprise integral skills, and training interventions commensurate with an introduction of new skilling objects or training interventions or changes to existing ones.

In accordance with the present invention, an interactive learning system and method are provided that identify a "gap" between the skill(s) possessed by the individual and those required by an organization and then creates a training regimen to fill that gap. The system and method precisely map the training to the defined required skills. In one aspect of the invention, an individualized learning sequence can be developed and presented to the user for the purpose of improving a user's skill and understanding with respect to selected subject matter. While known prior art systems use manual linking of recommended training elements with a desired skill to be learned, the present automatic invention reduces, by at least 95%, the time required to create a link.

Numerous other advantages and features of the present invention will become readily apparent from the following detailed description of the invention and the embodiments thereof, from the claims and from the accompanying drawings in which the details of the invention are fully and completely disclosed as a part of this specification.

Brief Description of the Drawings

FIG. 1A is a schematic diagram of an exemplary architecture for the system of the present invention.

FIG. 1B is a schematic diagram of a typical flow for a user through the functional components of the system of FIG. 1A.

FIG. 1C is a schematic diagram of the functional components of the system of FIG. 1A.

FIG. 1D is a flow diagram of the details of the matching process.

FIG. 1E is a flow diagram of the details of the gap analysis process.

FIG. 1F is a schematic diagram of a typical flow through the gap analysis process and a resultant recommendation.

FIG. 2 is a flow diagram of the method for creating an individual development plan.

FIGS. 3A and 3B illustrate in tabular fashion the data flow for creating a customized individual learning plan.

FIG. 4 is a schematic diagram of automatic matching, automatic linkage, and semiautomatic linkage for creating a training regimen.

FIG. 5 is a block diagram of the updating process.

FIG. 6 is a flow diagram of a process of selecting learning objects for the creation of a personalized learning experience.

Detailed Description of the Preferred Embodiments

A description of the preferred embodiments of the present invention will now be presented with reference to FIGS. 1–6.

FIGURE 1A is a schematic diagram of the hardware of a preferred embodiment of a system **10** to address the problem solved by the present invention; FIG. 1B is a

schematic diagram of a typical flow for a user through the functional components of the system; FIG. 1C is a schematic diagram of the functional components.

The system **10** incorporates a plurality of components (FIGS. 1A and 1C), including a control unit **12**, comprising, for example, a microprocessor. Associated with the microprocessor **12** are read-only memory **14** and read-write memory RAM **16**, which communicate with the processor **12** via a bidirectional bus structure **18**. The memories **14,16**, as is known in the art, can be used to permanently or temporarily store constants, programs being executed, or related data.

Also coupled to the bus **18** is a monitor, such as a CRT-type display **20**, disk drive **24**, CD-ROM drive **26**, and a user input device **28**. The user input device **28** may comprise, for example, a keyboard; it could alternately or in addition comprise a pointing device, such as a mouse, or touch screen.

A local area network (LAN) interface **30** enables the system **10** to communicate with the local network **31** (e.g., intranet). Further, a remote network interface **32** provides bidirectional communication with the World Wide Web or other remote networks (e.g., Internet), either through telecommunications or through other forms of data transmission without limitation.

In normal operation, software programs can be stored on and read from disk drive **24** or CD-ROM drive **26**. Information can also be read from the CD-ROM drive **26**. Both the disk drive **24** and the CD-ROM drive **26** can be used as sources from which databases can be accessed; alternatively, databases may be accessed via the LAN interface **30** or the remote network interface **32**.

During normal operation of the system **10**, visual information can be displayed on the display unit **20**, whether being used by a user or management personnel, under control of the processor **12**. Information can be read from disk drive **24** or CD-ROM drive **26** for the purpose of displaying same to the user/manager. A control program, or a suite of such programs, for the purpose of carrying out the processes of the present invention can be stored on one or both of disk drive **24** or CD-ROM **26** for access by processor **12**. Processor **12** is thus able to carry out any of a number of prestored processes that provide for the multiplicity of functions to be described herein.

A overall interactive schematic of one aspect of the invention, which comprises a computer-based method of automatically determining and providing an individualized learning sequence, is illustrated in FIG. 1B. In this embodiment of the method, an organization has identified a skill area in which it desires for a known individual to gain proficiency, and learning tools are available for addressing this need.

In the system **10**, the skills are assembled in a skills database **200**. When an individual is needed to perform what may be a new skill or an existing skill at a higher skill level, an analysis of the skill possessed already by the individual **204** is performed to determine whether a "gap" **206** exists between the skill **204** and that needed by the organization **202**. Preferably this analysis is performed electronically, with the use of gap generator software. The system **10**, through the use of gap analysis software of the present invention, creates a complete set of training recommendations to satisfy the gap. An individual development plan **208** for the given skill comprises these training recommendations.

The system's **10** gap analysis software (FIGS. 1E and 1F) converts the two incoming competency levels **80**, the user's current level and the desired level to be attained, to a decimal equivalent (percent) **81** (block **660**). This normalizes the values so that proper comparison may be made between different competency rating scales. These percentages are then compared with all training interventions linked to the desired skilling object (block **661**) in the training intervention metadata database **82**. Linked training interventions whose competency levels are within the range set by the incoming levels (i.e., \leq the desired level and \geq current level) are identified as the initial set of training recommendations **83** (block **662**).

A "complete" set of training recommendations **84** is accomplished by examining the prerequisite training intervention metadata element of each of the initial set of training recommendations. Prerequisites **85** are added to the initial set of training recommendations (block **663**), and are indented or highlighted to indicate that they are "prerequisites" and not the primary training recommendations required to fill a skill gap.

The information contained in each training recommendation comprises the training intervention ID and the competency level and the prerequisite training intervention IDs.

If a gap **206** exists, data representative of the linkage between the required skill and skill level (i.e., competency level) to the appropriate set of training interventions in the training intervention metadata database **207** are used to create an "individual development plan" (IDP) **208**. When desired, the course(s) to be taken from the regimen may be accessed from the course database **209**.

Typically, not only a particular skill will be required, but also a predetermined level of skill. To this end the skills database **200** will also comprise the individual's skill level for each skill. In this case, the "gap" **206** further comprises a difference in skill level between that which the individual possesses in that skill and that desired by the organization.

FIGURE 2 is a flow diagram of the method **40** for creating an individual development plan. In this aspect of the method the organization, typically by way of a supervisor or manager, identifies a skill and skill level desired to be possessed by the user **401**. The system **10** then accesses the skills database **402** and retrieves the user's skills and skill levels **403**. If the desired skill is not already possessed by user **404**, the desired skill is mapped with the appropriate training regimen **406**. If the desired skill is already possessed by the user **404**, the system then checks whether the desired skill level is less than or equal to the user's skill level **405**. If so, the system exits **408**. If the desired skill level is greater than that possessed by the user **405**, the desired skill is mapped with the appropriate training regimen **406**; that is, the skills domain is mapped to the training domain.

In some cases the gap **206** is sufficiently great that a plurality of courses will need to be assigned to the user. For example, if the user has a skill level of "1" for a particular skill, and the desired skill level is "5," the user will be assigned a training regimen comprising courses matched to skill levels 1 through 5 of that skill, so that all prerequisites are met en route to achieving the desired skill level.

Once the mapping has occurred **406**, the training regimen is presented to the user **407**. The types of information that may be obtained by the user is accessed via the

metadata resident in a third database **207**. The metadata comprise information that describes the attributes of a resource, the training regimen, and here serve as a link between the skills database and the mapping function of the system. Examples of metadata elements include, but are not intended to be limited to, an identifier (ID), a title of the training regimen, a description thereof, objectives therefor, duration, version, size, platform requirements, and prerequisites.

A tabular depiction of the data flow comprising the mapping is given in FIGS. 3A and 3B, wherein a first table **41** contains the user skills and organization requirements from databases **204,202**, including a column of skills **411**, the user's current skill level **412**, and the organization's desired skill level **413**. From these data is generated a second table **42**, demonstrating the "skills gap" for each skill, the first column **421** again comprising the skills, the second column **422** the user's current skill level, and the third column **423** the desired skill level. In this table **42**, the rows of table **41** have been deleted wherein the user's skill levels matched or exceeded those required.

In a third table **43** a training map is formed, wherein each skill **431** that appears in the second table **42** is located in the metadata database, and mapped to all available courses **433** and their respective skill levels **432**. Another important aspect of the invention is that a multiplicity of courses on the same skill having ascending skill levels may be defined hierarchically, so that an entry level is predefined for each course and a series of courses with ascending prerequisite skill levels included in the mapping. For example, in the table **44**, five levels of courses, AA, FA, FB, AB, and AC, are recommended to achieve the desired skill level of 5 for Knowledge 1.

It should also be noted that an important aspect of the present invention is that the source of each course **433** is immaterial, as is the skill level range, and that any number of courses, levels, and sources thereof are in principle accessible by the system, so long as the metadata have been entered into the third database so that they may be accessed by the system **10**.

As described previously, the metadata included in the metadata database **207** preferably include a "natural language" description **502** of each course. A schematic diagram of automatic matching, automatic linkage, and semiautomatic linkage (FIG. 4) indicates that, at this point in the process, an ID **501** of a skill in natural language in the individual's skills database **204** is automatically matched with a course description **502**, as well as other metadata elements, such as title **512**, also in a natural language, which is an element of the metadata set **500** for a course **510** (here Course AA) in the metadata database **207**. A matching algorithm **503** resident in a processor provides the exemplary results shown in block **504**. An important feature of the present invention is the ability for the administrator to select which training metadata element(s) and skilling object metadata element(s) the matching software examines, from one element to all elements. The matching algorithm **503** establishes probable matches between skills and training elements on the basis of probable relevancy, creating a match table with probability scores that provide a high level of certainty for making links of training elements to skills during the linking process.

To describe the matching process in more detail (FIG. 1D), the administrator performs a "setup" procedure, which comprises selecting metadata elements for each

course and skill that are to be considered, such as course title, course description, objectives, etc. (block 680). These selected “operating elements” (Table 1), such as title, description, discipline, etc., are taken from a table containing course information (Table 2) and skills (Table 3). The selected operating elements are then fed to a “harvest” program, which searches through all the words in the operating elements (block 681) and establishes a table (block 682) that contains a list of words found, the number of times each word occurred, and position of each word in the searchable operating elements. The positions of the words may be used during the matching step to give “bonus” points to matches that contain like phrases or word groupings. For example, “software programming” found in a training element and a skill element would provide bonus points in the match score.

Table 1		
SourceTable	SourceKeyCol	SourceValCol
Courses	CourseID	CourseTitle
Courses	CourseID	CourseOverview
Courses	CourseID	CourseObjectives
Skills	ObjID	Title

Table 2			
Course ID	Course Title	Course Overview	Course Objectives
12663	C++ Programming Fundamentals - Part 1	This is the first course in a two-part course covering the fundamentals of programming with the C++ programming language. This series will show you how to get the benefits of good software engineering and code reuse by using C++ and object-oriented programming techniques in business applications.	<ul style="list-style-type: none"> - Use C++ classes and functions. - Use C++ objects, data types, inline functions and referencing. - Use composition and association in C++ programs.

Table 3		
ObjID	SkillType	Title
KNOW1012	KNOW	Programming Languages:
TOOL412	TOOL	Oracle
KNOW581	KNOW	Television Programming
TOOL584	TOOL	UNIX (Solaris, BSD, etc.)

The administrator then runs a weighting program (block **685**), which compares the words in the word table with a word skip table, which contains a list of words that are to be ignored in the weighting process, such as "a," "an," "the," etc. If desired (block **683**), the administrator can modify the word skip table (block **684**) prior to running the weighting program (block **685**). The weighting program assigns a value to the word, shown as a "1" in Table 4, to those words that should be skipped during the matching process.

Table 4			
Id	Word	Count	Skip
354	c++	7	0
355	programming	6	0
382	and	5	1
361	the	4	1
392	use	3	1
364	in	3	1
394	functions	2	1
368	of	2	1
363	course	2	1
359	this	2	1
357	part	2	1
356	fundamentals	2	1
375	how	1	1
371	series	1	1
373	show	1	1
376	to	1	1
372	will	1	1
374	you	1	1
370	language	1	0
369	with	1	1
367	covering	1	1
365	a	1	1
379	good	1	1

Table 4			
Id	Word	Count	Skip
362	first	1	1
360	is	1	1
358	1	1	1
366	two	1	0
389	techniques	1	1
402	programs	1	0
401	association	1	0
400	composition	1	0
399	referencing	1	0
398	inline	1	0
397	types	1	1
396	data	1	1
395	objects	1	0
393	classes	1	0
377	get	1	1
390	business	1	1
378	benefits	1	1
388	oriented	1	0
387	object	1	0
386	using	1	1
385	by	1	1
384	reuse	1	0
383	code	1	0
381	engineering	1	0
380	software	1	0
403	languages	1	0
391	applications	1	1
415	oracle	1	0
416	television	1	0
417	unix	1	0
418	solaris	1	0
419	bsd	1	0
420	etc	1	1

The weighting program also creates a phrases table, Table 5, that contains two-, three-, and four-word phrases or word groupings. These phrases are used during the matching program to give “bonus” points to the total match score. For example, “software programming” in a training intervention element and a skilling object element would provide additional bonus points in the total match score.

Table 5			
id	Phrase	ObjID	Count
2435	business applications	12663	1
2431	c++ programming	12663	2
2433	c++ programming language	12663	1
2434	c++ programs	12663	1
2439	inline functions	12663	1
2441	language c++	KNOW1012	1
2436	object oriented	12663	1
2437	object oriented programming	12663	1
2438	oriented programming	12663	1
2443	programming language	KNOW1012	1
2444	programming language	12663	1
2442	programming language c++	KNOW1012	1
2440	television programming	KNOW581	1
2445	c++ objects	12663	1
2446	c++ class	12663	1

The weighting program further contains a function that allows the administrator to insert words into a word weight table and assign a weight (normally a value other than the default weight value of "1") to these words. This table, an example of which is found in Table 6, thereby allows the administrator to enhance the probability of valid matches due to known words in the training intervention metadata elements that are used by many different training interventions. For example, if the word "Microsoft" were used in describing many training interventions, the weight may be reduced to a value less than "1." The administrator would complete the word weight table before running the matching program.

Table 6		
ID	Word	Weight
14	microsoft	0.012
15	server	0.0096
16	network	0.024
17	system	0.047
18	software	0.275
19	information	0.092
20	management	0.046
21	application	0.089
22	window	0.0093
23	lotus	0.091

Another aspect of the present invention includes a system and method for creating the word skip and word weight tables. The guidelines are presented in Table 6A, along with examples for each type of word that is expected to be encountered. As a general rule, words in the word skip table have a point value of zero (0), while words in the word weight table can have any value. Words that are not placed in one of these tables are assigned a default weight of three (3), as indicated in the "Neither" column in Table 6A.

Table 6A				
If the word can be a:	For example	It goes in:		
		WordSkip Table	WordWeight Table with a point value as stated	Neither
1. Pronoun Only	He, she, it, they, I	X		
2. Preposition Only	For, at,	X		
3. Conjunction Only	And, but, for, nor	X		
4. Interjection Only	Yikes	X		
5. Determiner Only	The, an, a	X		

Table 6A				
If the word can be a:	For example	It goes in:		
		WordSkip Table	WordWeight Table with a point value as stated	Neither
6. Verb ONLY that are general and can apply to more than one subject	Perform, tell, organize	X Note: For those instances where the present participles are nouns e.g., burnishing), the word will follow guideline #22.		
7. Verb ONLY whose use is normally restricted to a single subject NOTE: Use carefully. Normally, follow the # 6 guideline; Verb ONLY above	Coagulate		X (All forms) Point Value = 5.0	
8. Verb and Adjective	Approximate	X [Present tenses ("s" or "es"), past and past participle ("ed"), present participle ("ing")]	X (Verb form and adjective) Point Value = .002	
9. Verb, Adjective, and Adverb	Direct	X a. [Present tenses ("s" or "es"), past and past participle ("ed"), present participle ("ing")] b. [Adverb]	X (Verb form, adjective, and adverb) Point Value = .001	

Table 6A

If the word can be a:	For example	It goes in:		
		WordSkip Table	WordWeight Table with a point value as stated	Neither
10. Verb and Noun	Strike, plan, punt, reserve, program, quarter	X [Past and past participle ("ed"), present participle ("ing")] Note: For those instances where the present participles are nouns (e.g., accounting), or adjectives/nouns (e.g., acting) the word will follow guideline #22.	X (Singular and plural forms of the noun) Point Value = .55	
11. Verb, Noun, and Adjective	Smash, plane, aggregate, associate	X [Past and past participle ("ed"), present participle ("ing")] Note: For those instances where the present participles are nouns (e.g., accounting), or adjectives/nouns the word will follow guideline #22.	X Singular and plural forms of the noun Point Value = .54	
12. Verb, Noun, Adjective and Adverb	Back	X [Past and past participle ("ed"), present participle ("ing")]	X Singular and plural forms of the noun Point Value = .53	
13. Verb, Noun, and Adverb	Bang	X [Past and past participle ("ed"), present participle ("ing")]	X Singular and plural forms of the noun Point Value = .52	

Table 6A				
If the word can be a:	For example	It goes in:		
		WordSkip Table	WordWeight Table with a point value as stated	Neither
14. Noun that has more than one definition and is not extremely general (see guideline #23) or unique (see guideline #15). Note: Included in these are the "...ness" nouns (e.g., accurateness) usually formed by adding "ness" to adjectives.	Administration, attention			X (Singular and plural forms of the noun)
15. Noun/Adjective that has ONLY one definition and is unique to a particular discipline or subject. If the word is used often in the training interventions, such as 'computer', it should not be given this weight. Instead, it should be given a weight of '3' (the default weight).	Pneumonia, hygrophytic, hygroscope, spacecraft		X (Singular and plural forms of the noun) Point Value = 6.0	
16. Noun and Adjective	Moderate, resident		X Singular form of the noun and adjective Point Value = .57 Plural form of the noun Point Value = 3.8	

Table 6A

If the word can be a:	For example	It goes in:		
		WordSkip Table	WordWeight Table with a point value as stated	Neither
17. Noun, Adjective, and Adverb	Andante, bimonthly		<p>X Singular form of the noun, adjective, and adverb</p> <p><u>Point Value = .56</u></p> <p>Plural form of the noun</p> <p>Point Value = 3.6</p>	
18. Adjective Only NOTE: See guideline #15 for unique adjectives..			<p>X</p> <p>Point Value = .004</p>	
19. Adverb Only		X		
20. Adjective and Adverb			<p>X</p> <p>Point Value = .003</p>	
21. Noun and Adverb	Tomorrow, sidesaddle		<p>X Singular form of the noun and adjective</p> <p><u>Point Value = .51</u></p> <p>Plural form of the noun</p> <p>Point Value = 3.4</p>	

Table 6A				
If the word can be a:	For example	It goes in:		
		WordSkip Table	WordWeight Table with a point value as stated	Neither
22. "ing" forms of a verb used as a noun. Note: If the 'ing' noun is very general (e.g., planning), use a point value of 0.501. Also, the use of the 0.501 value is used for 'ing' verbs that are not classified as nouns in the dictionary but sometimes intended to be nouns or adjectives by authors of skill or training intervention titles (e.g., delegating).	Accounting, beating, flying, sailing		X Singular form of the noun <u>Point Value = 2.3</u> Plural form of the noun Point Value = 4.0	
23. Nouns that are extremely general. This includes nouns that are also adverbs, verbs, and adjectives.	Concept, concepts, item, items, notion, notions, idea, ideas, basics, yes, fundamental, fundamentals		X Singular and plural forms of the noun Point Value = .0002	
24. Words that can be both a noun (singular and plural) and verb (present and 3 rd person present singular) that deal with a unique discipline or subject.	leister/ leisters microwave/microwaves, lullaby/lullabies		X Singular and plural forms of the noun Point Value = 5.5	
25. Very general abbreviations.	i.e., e.g.		Point Value = 0.0001	

Exceptions to these general rules are as follows:

If the word is a proper noun such as, for example, a company name that appears many times in the training intervention metadata elements (e.g., Microsoft, Lotus) in

conjunction with a product (e.g., Microsoft Excel) but only a few times in the skilling object metadata elements, then the word should be placed in the word weight table with a point value of "0.006." In cases such as these, the company name is viewed as an "adjective-like word" and, as such, is given a weight of close to the adjective weight (0.004) or 0.006. If the word is not a proper noun but is used often in the training intervention metadata, a weight of 0.05 is assigned.

If a word (or words) is used to describe a product/technology or an entity (e.g., Dolby, Visual Basic, Virtual Reality, and pneumonia), then the word (or words) should be placed in the word weight table. When assigning point values, caution should be exercised because common usage of these words will cause a higher than normal match scores for invalid associations between the skilling object and the training intervention. It is preferable that the administrator check the results of such action by running a harvest and then examine the results of such settings on the match scores. It may be necessary for the administrator to examine the words in the skilling objects. Adjustments (e.g., lowering of the point value or moving words between the word skip and word weight tables) may be necessary.

Typically, technical terms such as "Windows," "server," "Basic," or "Notes" that appear often in the training interventions' metadata are placed in the word weight table with a value of 0.50 and their singular/plural forms are assigned a weight according to the guidelines contained herein. Examples include: "Basic" = 0.50 and "basics" = 0.54; "Notes" = 0.50 and "note" = 0.55; "Access" = 0.50 and "accesses" = 0.55.

Unique words and abbreviations such as "Dolby," "PowerPoint," "NASA," or "pneumonia" would follow the guideline in Row 15 in Table 6A.

It will be understood by one of skill in the art that, although the exemplary words above primarily occur in the software and computer science arts, they are not intended to limit the breadth of the present invention, and that other subject areas will have their own terms of art for addition to the word skip and word weight tables.

Following the application of the weighting program, the match program is then utilized to perform a mathematical calculation on the tables (block 686) and create a match table (block 687) that shows which training interventions are matched to skilling objects with corresponding scores reflecting the probability of valid matches (Table 7). A score is calculated that, in an exemplary embodiment, is the sum of three scores. The higher the score, the higher the probability that a selected course supports the skill desired.

Score 1. An exemplary algorithm for calculating Score 1 is as follows, with the calculation performed for each word from word #1 to word #*n* in the desired skill and every course (i.e., training intervention). This score looks at each word in the skilling object and compares it to the words in each training intervention. If words are duplicated in the skilling object's words, the calculation is only performed once for each word. The software adjusts for singular and/or plurals. For example, a singular form of a word found in the training intervention and a plural form found in the skilling object are treated as equal. A default word weight value of "1" is given for each word unless modified by the word weight table.

$$\text{Score 1} = [(\text{count of word \#1 in course operating elements})(\text{weight of word \#1})] + [(\text{count of word \#2 in course operating elements})(\text{weight of word \#2})] + \dots + [(\text{count of word \#n in course operating elements})(\text{weight of word \#n})]$$

Score 2. For each combination of two, three, or four words in the skilling object's words that match two, three, or four word combinations in the training intervention's words,

bonus points (normally 500 for each match) are added to Score 2. The maximum score for Score 2 is set to 9000 (500 x 18 occurrences). The value of the bonus points for the two-, three-, or four-word phrase matches is adjustable by the administrator, but the maximum score for Score 2 cannot exceed 9000.

Score 3. Score 3 is obtained by comparing the words in the skilling object's and training intervention's "discipline" elements. An exact match places 10,000 points into Score 3. Score 3 may or may not be used, at the discretion of the administrator during the setup procedure. If the administrator knows that a common "discipline" element exists for some or all skilling objects and training interventions, he/she would select the elements that will be compared for establishing Score 3. If the element is selected for establishing Score 3, the element cannot be used in the Score 1 or Score 2 calculations.

In this method, Score 1 alone shows the relationship of words, without a high probability of the same contextual meaning between the skilling object and training intervention. A Score 1 value above 10 is typically a good indicator that a valid relationship exists.

A Score 2 value between 500 and 9000 indicates an increased probability of a valid relationship because of the combination of words or "like phrases." The higher the score for Score 2, the higher the probability of a valid relationship.

By adding Score 3 to Score 1 and Score 2, the probability of a valid relationship increases dramatically, because it tells us that the skilling object and the training intervention involve the same subject matter (e.g., mathematics, chemistry, information technology, leadership, etc.). For example, a total score of 19,053 would indicate a valid relationship of the skilling object and the training intervention. In this example, both the

skilling object and the training intervention could have indicated that the discipline is “IT Tools” (Score 3 = 10,000). There are at least 18 occurrences of matched two-word phrases such as “Microsoft Access” (Score 2 = 500 x 18 = 9000), and 53 occurrences of a word in the training intervention elements that match a word in the skilling object (Score 1 = 53).

Table 7 is an example of a Match Table showing a total score consisting of Score 1 + Score 2 and all word weights equal to “1.” This table is the result of running the matching software of the present invention on the operating elements of Table 1; the training intervention used is shown in Table 2; the skilling objects in Table 3; the results of the word skip table on the words table in Table 4, with the “1”s indicating that the words were not used in the calculation; and Table 5 showing the phrases.

Table 7			
ID	ObjID	CourseID	Score
41	KNOW1012	12663	511.0
42	KNOW581	12663	4.0

Returning to FIG. 4, the “score” **505** as calculated by the matching algorithm **503** for the results of the mapping of the skill ID **501** to the course description **502** is tabulated. A threshold level **506**, which is selectable by the administrator, provides a cutoff below which a course **502** is not considered. The administrator can also select matches “manually” and create desired links in the neighborhood of the threshold **506**. With automatic linkage **507** the system **10** proceeds directly to updating the appropriate course metadata element(s) **508**; with semiautomatic linkage **511** the administrator intervenes by setting a desired competency level **509** prior to updating the metadata element(s) **508**. The automatic

linkage **507** establishes the association of the skill to the course. The semiautomatic link **511** establishes the competency level acquired from the course, as well as the association.

Specifically, the administrator creates links along the semiautomatic pathway **511** by a combination of:

1. Selecting all matches for a range of scores (i.e., setting the lower and higher scores results in all matches equal to or greater than the lower score but less than or equal to the higher score to be linked) or above a selectable threshold level;
2. Selecting one match at a time (i.e., link the one match selected); or
3. Selecting multiple matches (i.e., highlighting various matches and then linking them all simultaneously).

The linking software of the present invention permits the administrator to view matches in a "linking screen," a portion of which is illustrated in Table 8.

Table 8					
id	ObjID	CourseID	Skilling Object Title	Training Intervention Title	Total Score
41	KNOW101 2	12663	Programming Languages: C++	C++ Programming Fundamentals - Part 1	511.0
42	KNOW581	12663	Television Programming	C++ Programming Fundamentals - Part 1	4.0

The administrator may view matches on the linking screen in a plurality of ways, including: by skilling object type; by skilling object identifier or group (e.g., "KNOW5**" displays all matches of skilling objects with IDs of "KNOW5xx"); by training intervention ID or group (e.g., "123**" displays all matches of training interventions with IDs of "123xx"); by match scores above or below a score; by exact match score or group (e.g., "5*" displays all scores between 50.0 and 59.9); all matches by skilling object or training intervention. The linking

software further permits sorting the information in any column, in ascending or descending order.

When the administrator makes a link, by selecting a "Link" button on the screen, the match is removed from the match table so that the administrator can continue the linking process on remaining matches if desired. If the administrator subsequently removes a link, the match information is restored into the match table.

When the linking process is complete, the metadata for each course are accessed, and the competency level of each skill linked to a course is set **509**, as stated above. The administrator then can call up a screen that permits viewing the training intervention metadata elements. When a particular training intervention is selected, the metadata are displayed along with all links to skilling objects. The administrator can then insert or modify the competency level for any linked training intervention. If the administrator wishes, he/she can also select a skilling object to see which training interventions are linked to it and insert or modify the competency level for any linked training intervention.

Returning to FIG. 3B, in a fourth table **44** a training plan is formulated, containing a column **441** of required skills and a column **442** of respective courses to be taken by the user. Thus it may be seen that the system precisely maps the training to the desired tasks, tools, and knowledge required by the organization. The table **44** is created dynamically, based on the user's gap information received by the gap analysis software and the links that were established during the linking process.

Another aspect of the present invention comprises a system and method for creating/updating the skilling object metadata database **204** and training metadata

database 207, including creating skilling object linkages within the training metadata database 207, which is an integral feature to providing an automated or semiautomated linking framework for creating individualized training regimens. Creating/updating the training metadata database 207 is accomplished by importing training intervention metadata from any desired source into an input means of the system 10 as is known in the art, or, in some cases, providing a linkage to an external source, such as via an intranet or internet.

In an automated linking framework the results of a matching algorithm are fed to a system 508 for updating the metadata database 207. In a semiautomated matching framework the results of the matching algorithm are passed to an intermediate step 509, wherein competency level(s) may be set prior to updating the metadata database 207. Also, setting a competency level may be accomplished anytime after links have been established through automatic linkage 507.

More specifically, the software adapted to perform updates when a user adds or changes a training intervention or a skilling object. This software reconciles the differences and focuses on the impact of the new or changed training intervention or skilling object on linkages and/or competency levels.

If the administrator initiates a transfer of revised information from an external source (e.g., a learning management system, human resource system, skills management system, CD-ROM, etc.), the update software compares the new information with previously archived information to determine if changes exist and allows the administrator to make

necessary updates to the linkages. This process (FIG. 5) comprises the following steps in an exemplary embodiment:

1. The administrator initiates an input of a new training intervention and skilling object element through the input/output interface (block **620**). If there is an existing archive (block **621**), the software queries the administrator as to whether it is desired to “compare” for differences or to “replace” (block **622**). “Replace” removes all matching and linking information in the database (block **623**).

2. If the administrator selects “compare” (block **622**), the software compares the new information in the current database with the archive (block **624**). If there are no differences, the software notifies the administrator (block **625**) and then branches to the “main menu” screen (block **626**). If there are differences, the software records the identifiers (IDs) of the objects (either new or modified objects) (block **627**) and creates a “change table” that lists the IDs of the subject objects (block **628**).

3. The software then performs a word harvest on the current database for the IDs contained in the change table (block **629**) and a word harvest on the archive for any IDs in the change table that exist in the archive (block **630**).

4. The software then compares the two word harvests and creates a “change word table” that reflects the differences in word counts, word positions, and phrases (block **631**).

5. After allowing the administrator the opportunity to view the change word table and to make any desired adjustments to the word skip or word weight tables, the software

then modifies the words table based on the differences in the change word table (block 632).

6. The software then runs the matching utility, and places the results in the match table (block 633).

7. The software then allows the administrator to view a screen that shows the changes by training intervention (Table 9) or skilling object (Table 10; block 634). The example shown in Tables 9 and 10 assumes that the information in training intervention "T1" and skilling object "C" has been modified and that there are two new skilling objects, "F" and "G."

Table 9. Training Intervention View							
Training Intervention ID	Training Intervention Title	Skilling Object ID	Skilling Object Title	Archive Match Score	New Match Score	Link Y/N/?	Competency Level
T1	Title 1	A	Title ABC	T1AA1	T1AA2	Y	51.5
		B	Title DEF	T1BB1	T1BB2	Y	30.0
		C	Title GHI	T1CC1	T1CC2	N	-
		F	Title PQR			?	-
T2	Title 2	C	Title GHI	T2CC1	T2CC2	Y	35.0
		D	Title JKL	T2DD1	T2DD2	Y	70.0
		E	Title MNO	T2EE1	T2EE2	Y	50.5
		F	Title PQR	--	T2FF2	?	-
T9	Title 3	F	Title PQR	-	T9FF2	?	-
		G	Title STU	-	T9GG2	?	-

Table 10. Skilling Object View							
Skilling Object ID	Skilling Object Title	Training Intervention ID	Training Intervention Title	Archive Match Score	New Match Score	Link Y/N/?	Competency Level
C	Title GHI	T1	Title 1	T1CC1	T1CC2	N	-
		T2	Title 2	T2CC1	T2CC2	Y	35.0
F	Title PQR	T1	Title 1	-	T1FF2	?	-
		T2	Title 2	-	T2FF2	?	-
		T9	Title 9	-	T9FF2	?	-
G	Title STU	T9	Title 9	-	T9GG2	?	-

8. Both views display every object affected by the changes, the titles of the object and training intervention, the archive match score, the new match score, any links that were previously made, and the corresponding competency levels. Additional features of these views include:

a. Modified and new objects are highlighted (here shown in boldface) to make it easy for the administrator to identify them from unchanged objects (block **635**).

b. Double clicking on an object (training intervention or skilling object) displays a window that shows the archive information and the new information (block **636**). This allows the administrator to make judgments on the impact of the new information on links. Obviously, new objects would not have previous information.

c. The provision to link (either by individual selections or between scores) and to set the competency levels provided (block **637**).

d. The link for each new or modified object is initially shown as a question mark (?), and its corresponding competency level is shown as a dash (-) (block **638**). This facilitates the administrator's being able to see which links and competency levels have to be made.

e. All links and competency levels for the changed objects should be made before exiting the update software (block **639**). If the administrator exits the update software without making a link and competency determination for every new object (i.e., changing the "Link" column to either "N" or "Y" and setting the

competency level for every Y), the software prompts the administrator (block 640), indicating that all links and competency levels for the changed objects were not made and that the working data will be saved to be worked on at a later time.

9. If the administrator exits the update software without completing the links and competency levels, the software provides a prompt to the administrator each time the program is entered.

FIGURE 1C represents the functional interplay between system elements for two anticipated accessors of the system 10: an administrator and a user/employee. The administrator's tasks include: (1) load/manage training metadata database; (2) load/manage skilling object metadata database; (3) maintain system configurations; (4) view training intervention metadata elements; (5) view skilling object metadata elements; (6) run software utilities; (7) link training interventions to skilling objects and set corresponding competency levels; and (8) prepare reports. The manager's tasks include: (1) view available courses; (2) view available skills; (3) search courses; (4) create/manage collections of courses based on existing objects; (5) match skills to courses; and (6) report progress. The user's tasks include: (1) select skills needed based on the gap analysis; (2) generate an individual development plan; and (3) send a curriculum to a course management system. The division of tasks within the system 10 are summarized in Table 11.

**Table 11. Precision Skilling Functions
Input/Output**

Function	The Administrator
1. Input Initial Skilling Object and Training Intervention Metadata Elements	Controls the transfer of the training intervention and skilling object metadata elements from the LMS to the PS database at the onset.
2. Input Updated Skilling Object and Training Intervention Metadata Elements	(See Update Software)
3. Output database tables to LMS	Controls the transfer of PS database tables to the LMS and/or removable media.
4. Provide reports	Creates the following reports: 1. Linked Skilling Objects – A listing of all linked skilling objects and the corresponding training interventions with competency levels. 2. Linked Training Interventions – A listing of all linked training interventions with competency levels and the corresponding skilling objects. 3. Unlinked Skilling Objects - A listing of all unlinked skilling objects. 4. Unlinked Training Interventions – A listing of all unlinked training interventions.

Function	The Employee
Accept Skilling Object Gap Information.	Via the LMS Gap Generator, initiates the sending of the gap information to PS.

Matching Software

Function	The Administrator
1. Setup Utility	Selects which skilling object and training intervention metadata elements will be used in the Harvest and Match programs.

Function	The Administrator
2. Harvest Utility	<p>Executes the Harvest Utility software that searches through all of the words in the training intervention and skilling object metadata operating elements to establish:</p> <ol style="list-style-type: none"> 1. The Words Table which contains: <ol style="list-style-type: none"> a) a listing of words found, and b) the number of times each word was found. 2. The juncWords_Courses Table that records the position of each word in the training intervention operating elements. 3. The juncWords_Skills Table that records the position of each word in the skilling object operating elements.
3. Weight Utility	<ol style="list-style-type: none"> 1. May modify the Word Skip Table prior to initiating the execution of the Weight Utility software. 2. Executes the Weight Utility software that: <ol style="list-style-type: none"> a) Compares the Word Table to the Word Skip Table. Words in the Words table that are found in the Word Skip Table are set to "1", indicating they will be skipped (i.e., ignored) during the Match Utility. b) Creates a Phrases Table that contains two and three word phrases or groupings. These phrases will be used during in the Matching Utility to give "bonus" points for phrase matches found in the training intervention and the skilling object. 3. Insert words and corresponding weights into the Word Weight Table to enhance the probability of valid matches due to known words in the training intervention metadata elements that are used by many training interventions. The word weights would normally be set to a value less than "1" (the default weight for each word).
4. Match	<p>Executes the Match Software that computes a score of probable valid match for each skilling object to training intervention (i.e., the training intervention supports the skilling object). The resulting Match Table provides a listing of probable valid matches with corresponding scores. The total match score is the sum of three individual scores (see Match Utility paragraph for details).</p>

Linking Software

Function	The Administrator
1. Link Training Intervention(s) to Skilling Object(s)	<p>Examines the contents of the Match Table via a Linking Screen to select which matches are valid matches and executes the Link Software that records the link to the skilling framework discipline in the training intervention metadata element (Classification.Purpose (Discipline).TaxonPath).</p> <p>The administrator makes links by combination of:</p> <ol style="list-style-type: none"> 1. selecting all matches between two match values, 2. selecting one match at a time (i.e., link the one match selected), or 3. selecting multiple matches (i.e., highlighting various matches and then linking them all at the same time). <p>Once a link is made, the match information for that link is removed from the Linking Screen.</p>
2. Setting Competency Levels	<ol style="list-style-type: none"> 1. Selects a screen where he/she can select: <ol style="list-style-type: none"> a) A single training intervention b) Grouping of training interventions (e.g., all 123xx courses) c) All training interventions 4. Then calls up the training intervention metadata for examination. 5. Enters the competency level for the link to each skilling object (i.e., at what competency level does this training intervention support each of the linked skilling objects).
3. Change Competency Levels	<ol style="list-style-type: none"> 1. Selects a screen where he/she can select: <ol style="list-style-type: none"> a) A single training intervention b) Grouping of training interventions (e.g., all 123xx courses) c) All training interventions 4. Then calls up the training intervention metadata for examination. 5. Changes the competency level for any link to a skilling object (i.e., at what competency level does this training intervention support each of the linked skilling objects).
4. Delete a Link	<ol style="list-style-type: none"> 1. Selects a screen where he/she can select: <ol style="list-style-type: none"> a) A single training intervention b) Grouping of training interventions (e.g., all 123xx courses) c) All training interventions 2. Then calls up the training intervention metadata for examination. 3. Deletes any link to a skilling object. <p>Note: Any link that is deleted results in having the original match information placed back into the Linking Screen.</p>

Update Software

Function	The Administrator
1. Input Updated Skilling Object and Training Intervention Metadata Elements	<ol style="list-style-type: none"> 1. Initiates the input of the new training intervention and skilling object elements through the Input/Output Interface. 2. If there is an existing archive (i.e., Precision Skilling has previously operated on the elements), the Update Software queries the administrator as to whether he/she wants to "Compare" for differences or to "Replace". "Replace" removes all matching and linking information in the current database. NOTE: Replacement is not normally done in an operational environment. 3. If the administrator selects "Replace", the software branches to the 'Main Menu' screen. 4. If the administrator selects 'Compare', the software compares the new information in the current database with the archive. <ol style="list-style-type: none"> a) If there are no differences, the software notifies the administrator and then branches to the 'Main Menu' screen b) If there are differences, the software records the identifiers (ID's) of the objects (either new or modified objects) and creates a "Change Table" that lists the ID's of the subject objects. <ol style="list-style-type: none"> 1) The software then performs a word harvest on the current database for the ID's contained in the Change Table and a word harvest on the archive for any ID's in the Change Table that exist in the archive. 2) The software then compares the two harvests and creates a "Change Word Table" that reflects the differences in word counts. 3) The software pauses to allow the administrator the opportunity to view the Change Word Table and to make any adjustments to the Word Skip or Word Weight tables
2. Adjust Word Skip and Word Weight Tables	The administrator views the Change Word Table and to make any adjustments to the Word Skip or Word Weight tables.

Function	The Administrator
3. Access impact of changes on link information	<ol style="list-style-type: none"> 1. When finished with any changes he/she might make, the administrator resumes the Update software. <ol style="list-style-type: none"> a) The software then modifies the "Words Table" based on the differences in the Change Word Table. b) The software then runs the Match Utility on the changed training intervention and skilling object metadata databases, placing the results in the Match Table. c) The software then allows the administrator to view a screen that shows the changes by training interventions or skilling objects. Both views display every object affected by the changes, the titles of the skilling object and training intervention, the archive match score, the new match score, any links that were previously made, and the corresponding competency levels. Modified and new objects are highlighted to make it easy for the administrator to identify them from unchanged objects. 2. The administrator then selects (e.g., double clicking on the object) an object (training intervention or skilling object) to have the software display a window that shows the archive information and the new information. This allows the administrator to make judgements on the impact of the new information on links. Obviously, new objects would not have previous information.
4. Update Link Information	<p>If the administrator determines changes are needed to the link information or there are new objects (training interventions or skilling objects), he/she inserts the appropriate information.</p> <ol style="list-style-type: none"> 1. The software provides the capability for the administrator to link (by individual selections, between scores, or above a threshold score) and to set the competency levels. 2. The software initially shows the link for each new or modified object as a question mark (?) and its corresponding competency level as a dash (-). This makes it easy for the administrator to see which links and competency levels he/she has to make. <p>Note 1: All links and competency levels for the changed objects should be made before the administrator exits the Update Software. If the administrator exits the Update Software without making a link and competency determination for every new object (i.e., changing the Link column to either N or Y and setting the competency level for every Y), the software prompts the administrator indicating that all links and competency levels for the changed objects were not made and that the working data will be saved to be worked on at a later time.</p> <p>Note 2: If the administrator exits the Update Software without completing the links and competency levels, the software provides a prompt to the administrator each time he/she enters Precision Skilling.</p>

Gap Analysis Software

Function	The Software
1. Accept Gap Information	Receives information for one skilling object gap
2. Convert Gap Levels to Percentages	The software converts the levels to percentages only if the gap levels are not in percentages. Note: The administrator establishes whether to convert or not convert based on the client's rating scales. It may be possible for a client who is supporting several skill frameworks to have some gaps generated as percentage and others as simple rating scales (with decimals).
3. Compile the List of Training Interventions	<ol style="list-style-type: none"> 1. The software examines the Skilling Object ID to compile a list of linked training interventions with competency levels that fall within the range of the "desired" level and the "actual" level. 2. For every training intervention identified in the list, the software looks at the Training Intervention "Prerequisite" element and adds the prerequisites to the list.
4. Send the Training Interventions to the LMS	<p>The software sends an ordered list of training interventions to the LMS. Each training intervention in the list includes:</p> <ol style="list-style-type: none"> 1. The Training Intervention ID 2. The Competency Level of the training intervention for the skilling object expressed in the form received by the LMS (i.e., single digit, with or without fraction, or percentage). 3. Followed by the Prerequisite IDs

The administrator's interactions include a login **701** to the processor **12**, which typically includes an entry of a password. The administrator is permitted by the system to access the user management subsystem **702** and a user database **703** thereby.

The administrator is responsible for automatically loading the desired skilling object metadata elements into the skilling object metadata database **200**, via a routine **704** adapted therefor. The administrator is also responsible for automatically loading training intervention metadata elements into the training intervention metadata database **207** via a routine **706** adapted therefor. Help files **707** are also provided for guiding the

administrator through the system. For courses that do not contain assessment features therewith, the administrator can use an assessment architect **708** to create them.

The administrator uses the matching routine **503** to match training interventions in the training intervention metadata database **207** to skilling objects in the skilling object metadata database **200**. The matching routine **503** creates a Match Table **504** that contains scores for each match; the higher the score, the higher the probability of a valid association. Help files **709** are also provided here. The linking software **711** establishes a link and competency table (actually a subset of the training intervention metadata database) that reflects the valid association of skilling objects to training interventions. These associations and corresponding competency levels are used during the gap analysis process **202-207**.

The user/employee accesses the system **10** and initiates the gap analysis process **202 - 207**, as described earlier. The gap analysis software establishes an individual development plan **208** for the incoming skill gap. The system **10** exports the individual development plan **208** to an external learning management system **710**, which the user accesses to take or register for the prescribed training interventions (courses).

From the foregoing, it will be observed that numerous variations and modifications may be effected without departing from the spirit and scope of the invention. It is to be understood that no limitation with respect to the specific apparatus illustrated herein is intended or should be inferred. It is, of course, intended to cover by the appended claims all such modifications as fall within the scope of the claims.